応用線形代数　最終レポート

B243343
高岡俊輔

1. 計算精度評価

計算精度を確認するために教科書内の問題をいくつか解き確認した。
解いた問題とその答えを以下に示す。
行列式　教科書p49の例1

$$\begin{vmatrix} 2 & 3 & 0 \\ 0 & 2 & 1 \\ 5 & -4 & -4 \end{vmatrix} = 7$$

連立方程式　教科書p26

$$\begin{cases} 2x_2 - 3x_3 = -4 \\ x_1 - 2x_2 + 3x_3 = 2 \\ 3x_1 - 8x_2 + 6x_3 = 0 \end{cases} \qquad x_1 = -4, x_2 = 0, x_3 = 2$$

固有値と固有ベクトル 教科書p135　問4(2)

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & -3 \\ 0 & -2 & 1 \end{bmatrix}$$

固有値1 : 1 , ベクトル[1,0,0]
固有値2 : 4 , ベクトル[1,-3,3] = [-0.22…, 0.68…, -0.68..]
固有値3 : -2 , ベクトル[-5,3,3]= [-0.76…, 0.45…, 0.45..]

プログラムの結果を以下に示す。本課題において、使用した言語はpythonであり,numpyライブラリを用いてそれぞれ計算を行った。
行列式　教科書p49の例1

```
matrix = [[2,3,0],
          [0,2,1],
          [5,-4,-4]
                  ]


print(linalg.det(np.array(matrix)))#行列式計算
```

✓ 0.0s

```
7.0
```

連立方程式　教科書p26

```
#教科書p26
left_side = [
    [0,2,-2],
    [1,-2,3],
    [3,-8,6]
    ]
right_side = [-4,2,0]

print(solve_linear_equation(left_side, right_side))
```

✓ 0.0s

```
[-4.  0.  2.]
```

固有値と固有ベクトル 教科書p135　問4(2)

```
#教科書p135
n_array = np.array([[1, 2, 3],
 [0, 1, -3],
[0, -3, 1]])


v,w = np.linalg.eig(n_array)
print(v)
#固有ベクトルを表示する
print(w)
```

✓ 0.0s

```
[ 1.  4. -2.]
[[ 1.          -0.22941573 -0.76249285]
 [ 0.           0.6882472   0.45749571]
 [ 0.          -0.6882472   0.45749571]]
```

結果を見比べてみると、行列式と連立方程式は正しく計算できていることが分かる。固有値と固有ベクトルについて、見てみると固有値はあっているがそれに対応する固有ベクトルがおかしいことが分かる.。これは、固有ベクトルが単位ベクトル化して出力されるため数値が小数になっているためである。元の比に戻すとその値は正しいことが分かった。

2. 計算時間評価

この課題を進めていくうえで複数回の計測の結果、計算時間にはばらつきがあることが分かったため、30回計測を繰り返し、その平均値の近似曲線をもとめた。

2.1 計算量を仮定した近似

最初に、アルゴリズムの計算量からデータを近似する曲線を考えた。行列式を解くために使用したアルゴリズムは、LU分解であり、これは計算量がO(N^3)であるため、3次多項式によって線形回帰を行った。図1は、1から1000次までの行列式をとく時間の平均の推移と、その近似曲線を示す。

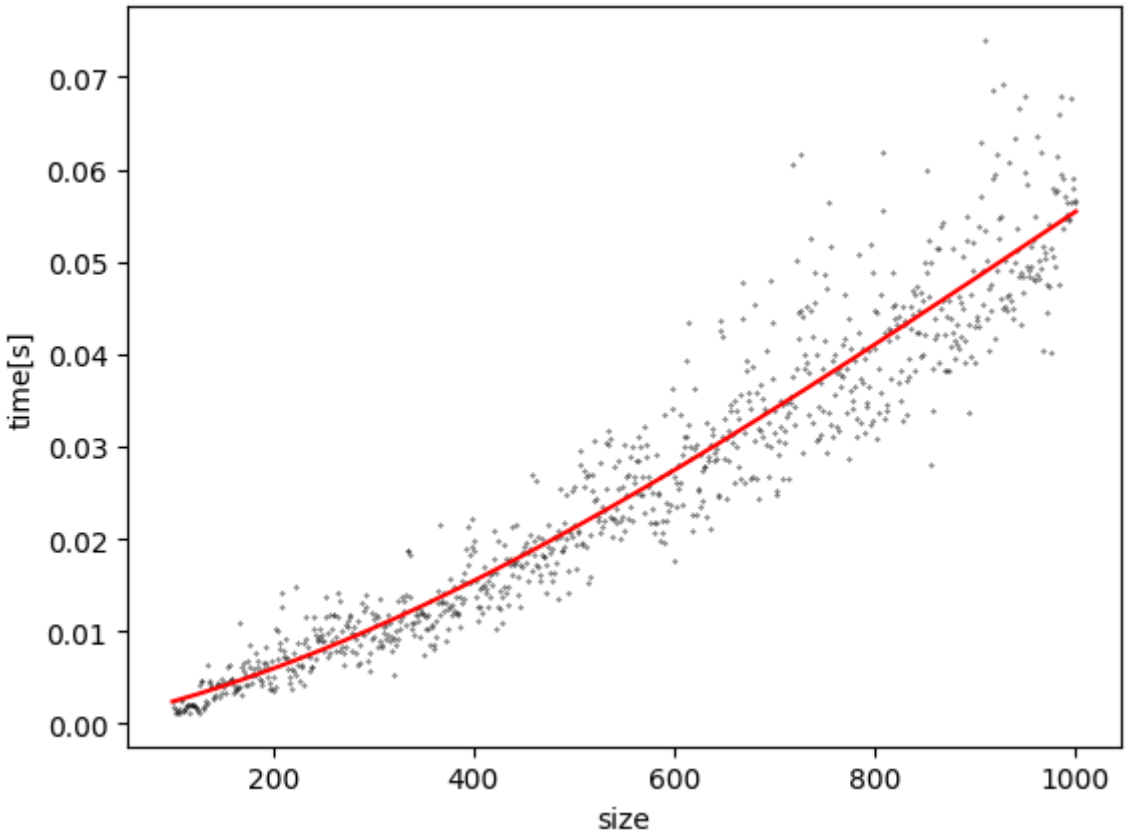

図1 行列式の計算時間の計測結果

図1の多項式回帰の結果を以下に示す。
多項式:y = 0.00000e+00 + 2.11115e-05 * x + 5.23591e-08 * x^2 + -1.78170e-11 * x^3
決定係数R^2 : 0.9130442131923542

決定係数は1に近いほどその多項式がもとのデータを説明できていることを示す。そのため、この多項式はかなりデータを表すことができていることが分かる。

次に、N次の連立方程式の計算時間を調査した。同じように30回計測してその平均値の近似曲線を求めた。連立方程式を解くために使用したアルゴリズムは、ガウスの消去法であり、これの計算量はO(N^3)であるため、3次多項式によって線形回帰を行った。図2は、1から1000次までの計算時間の平均の推移と近似曲線を示す。

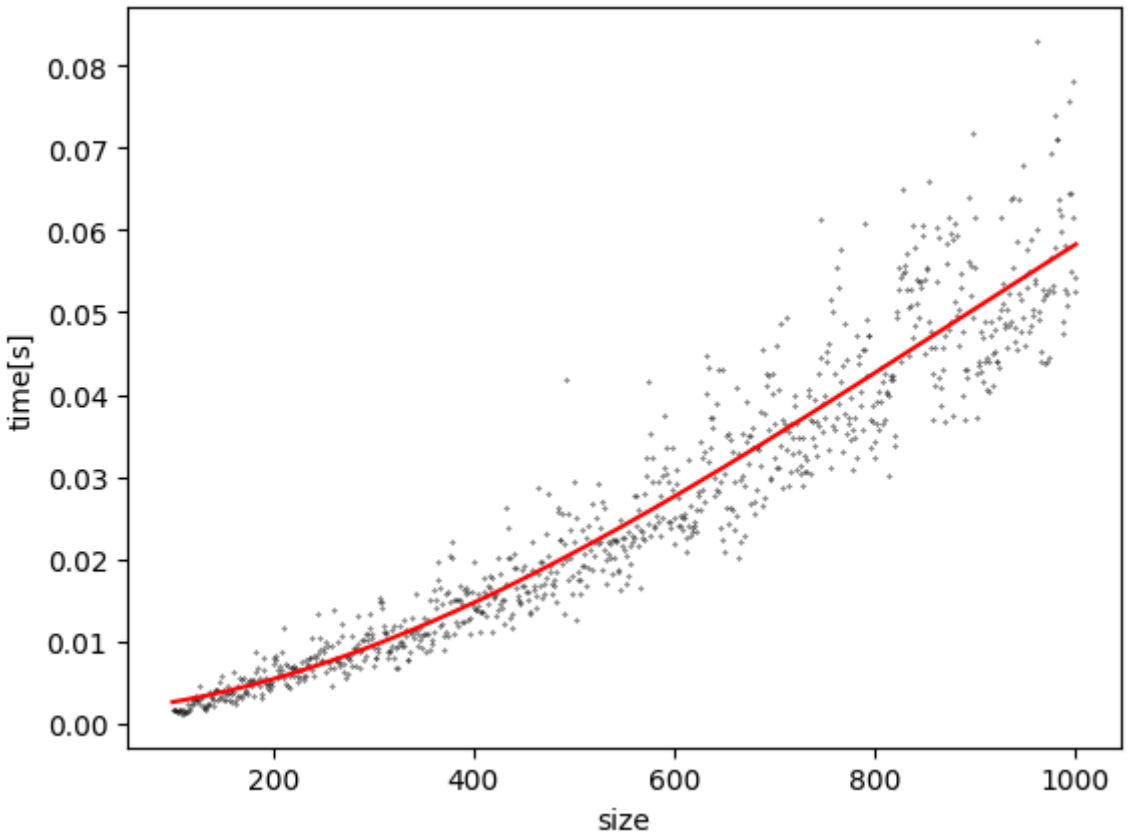

図2 連立方程式の計算時間の計測結果

図2の多項式回帰の結果を以下に示す。
多項式:y = 0.00000e+00 + 4.95783e-06 * x + 8.39890e-08 * x^2 + -3.19790e-11 * x^3
決定係数R^2 : 0.9100643642134882

決定係数は1に近いほどその多項式がもとのデータを説明できていることを示す。そのため、この多項式はかなりデータを表すことができていることが分かる。

次に、N次の行列の固有値と固有ベクトルの計算時間を調査した。同じように30回計測してその平均値の近似曲線を求めた。固有値と固有ベクトルを解くために使用したアルゴリズムは、QR法であり、これの計算量はO(N^3)であるため、3次多項式によって線形回帰を行った。図3は、1から1000次までの計算時間の平均の推移と近似曲線を示す。

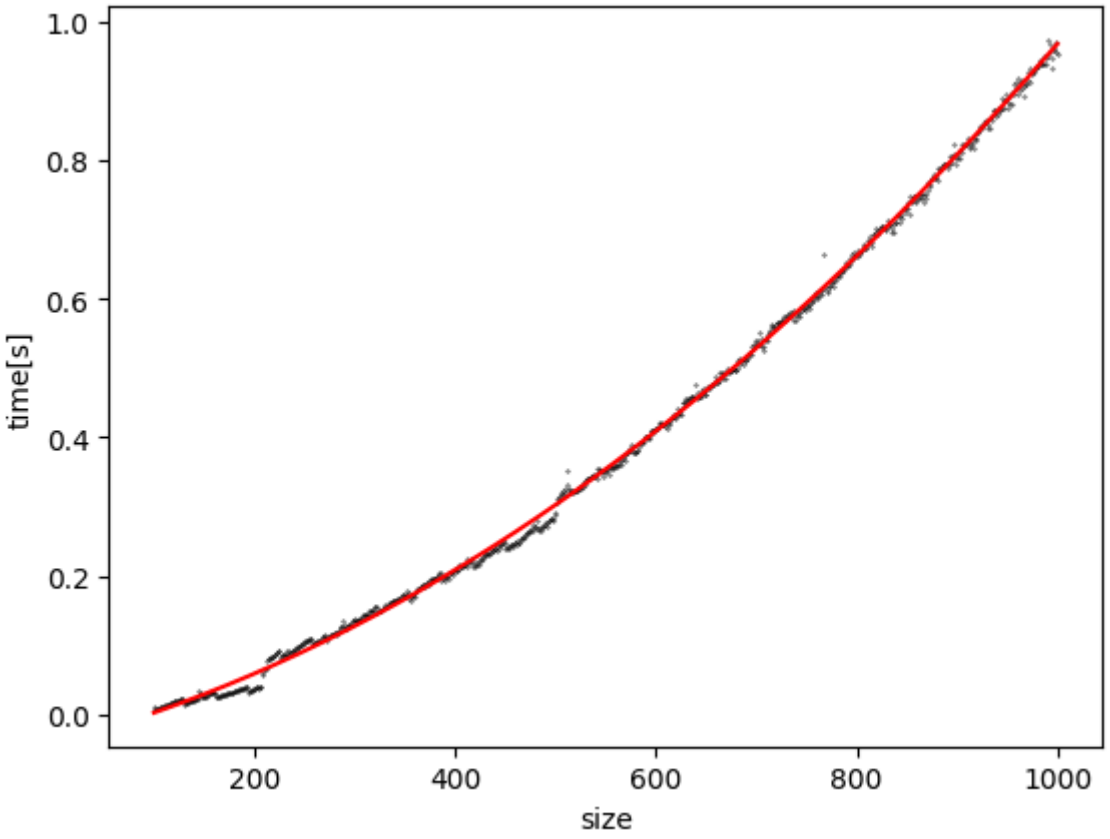

図3　行列の固有値と固有ベクトルの計算時間の計測結果

図3の多項式回帰の結果を以下に示す。
多項式:y = y = 0.00000e+00 + 3.82480e-04 * x + 5.94020e-07 * x^2 + 3.19751e-11 * x^3
決定係数R^2 : 0.9991068487373024

決定係数は1に近いほどその多項式がもとのデータを説明できていることを示す。そのため、この多項式はかなりデータを表すことができていることが分かる。

2.2　y=ax^n + bによるフィッティング

次に、記録されたデータをy = ax^n + bの形でフィッティングすることで、計算が何乗に比例するデータなのかを確認した。それぞれの近似結果を図4、図5、図6に示す。
フィッティング結果から、それぞれ以下のようにまとめられる。

1. 行列式の計算は、次数の1.39乗
2. 連立方程式の計算は、次数の1.50乗
3. 行列の固有値と固有ベクトルの計算は、次数の1.69乗

にそれぞれ比例することわかった。これは、O(N^3)よりも計算量が早いことを示す。

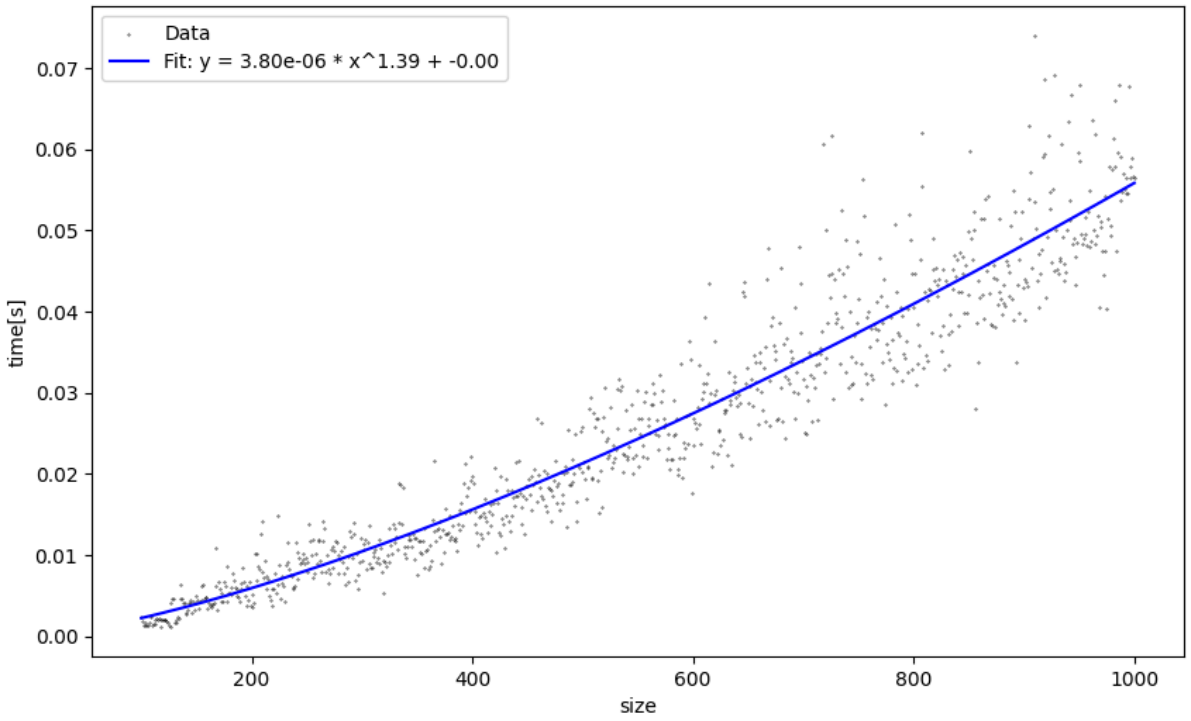

図4 行列式の計算時間の計測結果

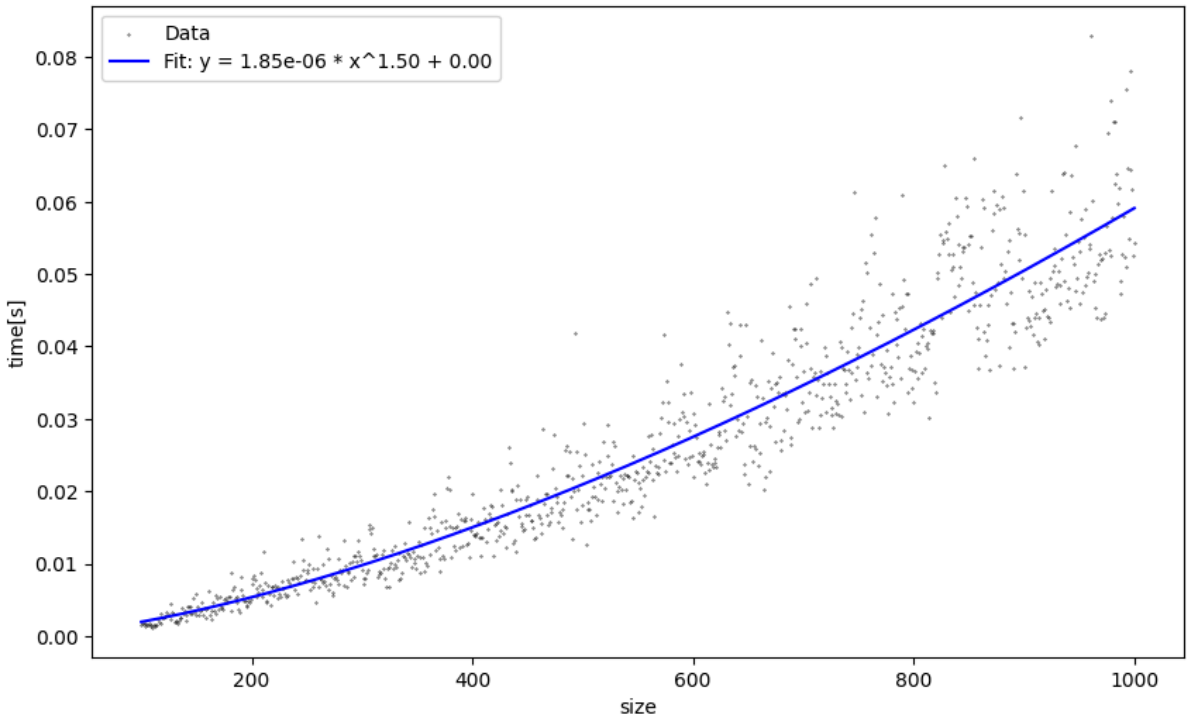

図5 連立方程式の計算時間の計測結果

図6 行列の固有値と固有ベクトルの計算時間の計測結果

3. まとめ

本課題では、pythonのnumpyライブラリを用いてそれぞれの計算を行いその計算時間の推移の近似曲線を3次多項式回帰によって求めた。これにより、3次多項式はかなり近似できることがわかった。次に、y=ax^n + b の形でフィッティングすることでデータが実際に次数の何乗に比例するのかを確認した。これによって、行列式の計算は次数の1.39乗、連立方程式の計算は次数の1.50乗、行列の固有値と固有ベクトルの計算は次数の1.69乗に比例することがわかった。今回の結果は、各種計算の実際の計算時間が理論的な計算量よりも低いことがわかる。